Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Theoretical Computer Science 304 (2003) 443–447

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Note

The approximate period problem for DNA alphabet

V.Y. Popov

Department of Mathematics and Mechanics, Ural State University, 620083 Ekaterinburg, Russia

Received 19 August 2002; accepted 25 February 2003
Communicated by A. Salomaa

Abstract

We show that the approximate period problem for DNA alphabet is **NP**-complete.

Keywords: Periodicity; Approximate periods; Distance function; **NP**-complete

Regularities in experimentally obtained data often reveal important knowledge about the underlying physical system. Regularities in a biological sequence can be used to identify the sequence among other sequences, or to infer information about the evolution of the sequence. The genomes of eukaryotes, i.e. higher-order organisms such as humans, contain many regularities. Tandem repeats, or tandem arrays, which are consecutive occurrences of the same string, are the most frequent. For example, the six nucleotides *TTAGGG* appear at the end of every human chromosome in tandem arrays that contain between one and 2000 copies [3]. Finding occurrences of repeated substrings in string is a widely studied problem. In biological sequence analysis searching for tandem repeats is used to reveal structural and functional information. However, searching for exact tandem repeats can be too restrictive because of sequencing and other experimental errors. A natural extension of the repetition is to allow errors.

Traditionally, the alignment notation has been used to illustrate a comparison between two or more sequences. Given a set of strings $X = \{x_1, x_2, \ldots, x_k\}$ on an alphabet, Σ, a multiple alignment of X is a set of strings

$$A = \{a_1, a_2, \ldots, a_k\},$$

E-mail address: vladimir.popov@usu.ru (V.Y. Popov).

0304-3975/03/$ - see front matter
doi:10.1016/S0304-3975(03)00211-1

$|a_1| = |a_2| = \cdots |a_k| = n$, $n \geqslant |x_i|$ for $1 \leqslant i \leqslant k$, on augmented alphabet $\Gamma = \Sigma \cup \{\Delta\}$ such that each string a_i is a copy of x_i into which $n - |x_i|$ copies of special symbol Δ have been inserted. Symbol Δ is called an indel and represents the insertion or deletion of a particular symbol in one string relative to another [4].

A penalty matrix specifies the substitution cost for each pair of characters and the insertion/deletion cost for each character. The weighted edit distance between x and y is the minimum cost to convert x to y using a penalty matrix.

How we consider an example of multiple alignment and costs. Let $U \leftrightarrow V$ costs 1, $U \leftrightarrow \Delta$ costs 2, $\Delta \leftrightarrow \Delta$ costs 1 where $U, V \in \{A, C\}$. Let $s_1 = ACACACAC$, $s_2 = CACACACAA$. Denote by $\delta(s_1, s_2)$ the cost to convert s_1 to s_2. If we consider the following alignment of s_1 and s_2:

$$ACACACAC\Delta,$$

$$CACACACAA,$$

then $\delta(s_1, s_2) = 10$. If we consider the following alignment of s_1 and s_2:

$$\Delta ACACACAC,$$

$$CACACACAA,$$

then $\delta(s_1, s_2) = 3$.

We consider the notion of approximate periods. This notion is first discussed in [5]. Let δ be an edit distance function specified by a penalty matrix. Given a strings x, a distance function δ, and an integer t, we define the t-approximate period u of x as follows. Let ε denotes the empty string. If there exists a partition of x into disjoint blocks of substrings, i.e., $x = p_1 \cdots p_r$, $p_i \neq \varepsilon$, such that $\delta(u, p_i) \leqslant t$ for $1 \leqslant i < r$, and $\delta(u', p_r) \leqslant t$ where u' is some prefix of u, then u is a t-approximate period of x (see [5]).

Consider the following problem:

The approximate period problem (AP)

Instance: A finite alphabet Γ, a string x from Γ^*, a penalty matrix M, and a positive integer t.

Question: Is there a string $u \in \Gamma^*$ such that u is a t-approximate period of x?

It is shown in [5] that if $|\Gamma| \geqslant 9$ then the AP problem is **NP**-complete. Since the alphabet for biological sequences is often of fixed constant size, e.g. DNA sequences and RNA primary sequences have alphabet of size 4, it is of interest to consider AP problem for $\Gamma_{\mathrm{DNA}} = \{A, G, C, T, \Delta\}$ where Δ is an indel, and $\{A, G, C, T\}$ is the natural DNA alphabet.

Theorem. *The AP problem for* Γ_{DNA} *is* **NP**-*complete*.

Proof. It is easy to see that the AP problem is in **NP**.

Let $|x| = |y|$. The Hamming distance between x and y is the smallest number of letter substitutions that convert x to y. Let y_1 and y_2 be finite strings. Let $d(y_1, y_2)$ denote the Hamming distance between y_1 and y_2. We consider the following problem:

	A	G	C	T	Δ
A	0	m	d	$t+1$	$t+1$
G	m	0	d	$t+1$	$t+1$
C	d	d	$2d$	$t+1$	$t+1$
T	$t+1$	$t+1$	$t+1$	0	$t+1$
Δ	$t+1$	$t+1$	$t+1$	$t+1$	0

Fig. 1. The penalty matrix M.

The decision version of the closest string problem (CS)

Instance: A finite alphabet Σ, a set $S=\{s_1,s_2,\ldots,s_n\}$ of strings, $S\subseteq\Sigma^m$, and a positive integer d.

Question: Is there a string s of length m such that for every string $s_i\in S$, $d(s,s_i)\leqslant d$?

The CS problem is **NP**-complete even restricting to a binary alphabet [1,2]. We will assume that $\Sigma=\{A,G\}$. Now we transform an instance of the CS problem into an instance of the AP problem as follows.

- $\Gamma=\Sigma\cup\{C,T,\Delta\}=\Gamma_{\mathrm{DNA}}$,
- $x=WTs_1TW^2Ts_2TW^3Ts_3T\cdots W^nTs_nTW^{n+1}$ where $W=TC^mT$,
- $t=md$,
- define the penalty matrix M as in Fig. 1.

It is easy to see that this transformation can be done in polynomial time. Note that the resulting distance is not a metric because $\delta(C,C)\neq 0$.

Let us show that if there is a string u such that u is a t-approximate period of x, then $u=Tu'T$ where u' is a string in alphabet $\{A,G\}$.

Let us consider a partition of x into disjoint blocks of substrings: $x=p_1\cdots p_r$. By definition $\delta(T,V)=\delta(V,T)=t+1$ where $V\in\{A,G,C,\Delta\}$. If u is a t-approximate period of x, then $\delta(u,p_i)\leqslant t$ for $1\leqslant i<r$, and $\delta(u',p_r)\leqslant t$ where u' is some prefix of u. Thus, to each T in u there must be a T in p_j and to each T in p_j there must be a T in u, $1\leqslant j\leqslant r$.

First, suppose that u has no T. Clearly, there exists a partition block of x which has at least one T, and the distance between u and the partition block is greater then t. Therefore, u must have at least one T. Suppose that u has no more than one T. In this case, $u=u'Tu''$ where $u',u''\in\{A,G,C,\Delta\}^*$. Consider the alignment of $p_1,\ldots,p_r$ induced by u. Let $\overline{p_i}$ be the supersequence of p_i induced by the alignment where $1\leqslant i\leqslant r$. It is easy to see that $\overline{p_1}=p_1'Tp_1''$ where $p_1'\in\{\Delta\}^*$, $p_1''\in\{C,\Delta\}^*$. It is clear that either $\delta(u,\overline{p_1})>t$ or

$$\delta(u',p_1')+\delta(u'',p_1'')\leqslant t. \tag{1}$$

Since u is a t-approximate period of x, $\delta(u,\overline{p_1})\leqslant t$. Thus, in view of (1),

$$\delta(u',p_1')\leqslant t \tag{2}$$

and

$$\delta(u'', p_1'') \leqslant t. \tag{3}$$

By (2) it is easy to see that

$$u' \in \{\Delta\}^*. \tag{4}$$

Since $\overline{p_1} = p_1' T p_1''$, $\overline{p_2} = p_2' T p_2''$ where $p_2' \in \{C, \Delta\}^*$, $p_2'' \in \{\Delta\}^*$. It is clear that either $\delta(u, p_2) > t$ or

$$\delta(u', p_2') + \delta(u'', p_2'') \leqslant t. \tag{5}$$

Since u is a t-approximate period of x, $\delta(u, p_2) \leqslant t$. Therefore, by (5)

$$\delta(u', p_2') \leqslant t \tag{6}$$

and

$$\delta(u'', p_2'') \leqslant t. \tag{7}$$

In view of (3) and (6), $p_2' \in \{\Delta\}^*$. Similarly, by (7)

$$u'' \in \{\Delta\}^*, \quad p_1'' \in \{\Delta\}^*.$$

Since C^m is a subsequence of $p_1'' p_2'$, $p_1'' \in \{\Delta\}^*$, and $p_2' \in \{\Delta\}^*$, u must have at least two T.

Now suppose that for some integer k

$$u = u' T u'' T \cdots u^{(k)} T u^{(k+1)},$$

where $u^{(i)} \in \{A, C, G, \Delta\}^*$ for all i. It is easy to see that in this case

$$\overline{p_1} = p_1' T p_1'' T \cdots p_1^{(k)} T p_1^{(k+1)},$$

where $p_1^{(i)} \in \{A, G, C, \Delta\}^*$ for all i. Since u is a t-approximate period of x, $\delta(u, \overline{p_1}) \leqslant t$. Therefore,

$$\delta(u', p_1') + \delta(u'', p_1'') + \cdots + \delta(u^{(k)}, p_1^{(k)}) + \delta(u^{(k+1)}, p_1^{(k+1)}) \leqslant t. \tag{8}$$

By (8)

$$\begin{aligned}
&\delta(u', p_1') \leqslant t,\\
&\delta(u'', p_1'') \leqslant t,\\
&\quad \cdots\\
&\delta(u^{(k)}, p_1^{(k)}) \leqslant t,\\
&\delta(u^{(k+1)}, p_1^{(k+1)}) \leqslant t.
\end{aligned} \tag{9}$$

Suppose that $k = 2p + 1$ where $p \geqslant 1$. In this case

$$\overline{p_2} = p_2' T p_2'' T \cdots p_2^{(k)} T p_2^{(k+1)},$$

where $p_2'' \in \{\Delta\}^*$. Clearly, $p_1'' \in \{C, \Delta\}^*$, and C^m is a subsequence of p_1''. In view of (9), it is easy to check that $u'' \notin \{\Delta\}^*$. Since $\delta(u, \overline{p_2}) \leqslant t$,

$$\delta(u', p_2') + \delta(u'', p_2'') + \cdots + \delta(u^{(k)}, p_2^{(k)}) + \delta(u^{(k+1)}, p_2^{(k+1)}) \leqslant t. \tag{10}$$

In view of (10), $\delta(u'', p_2'') \leqslant t$. Since $p_2'' \in \{\Delta\}^*$, $u'' \in \{\Delta\}^*$. Therefore, $k \neq 2p+1$. Suppose that $k = 2p$ where $p \geqslant 1$. It is not hard to check that in this case

$$\overline{p_1} = (W)_\Delta T(s_1)_\Delta T(W^2)_\Delta T(s_2)_\Delta T(W^3)_\Delta T(s_3)_\Delta T \cdots (W^l)_\Delta T(s_l)_\Delta T(W^r)_\Delta,$$

where

$$(Y)_\Delta = \Delta^{q_1} Y_1 \Delta^{q_2} Y_2 \cdots \Delta^{q_b} Y_b \Delta^{q_{b+1}},$$

$$Y = Y_1 Y_2 \cdots Y_b,$$

$b \geqslant 1$, $q_i \geqslant 0$ for all i. Clearly, C^m is a subsequence of $(W)_\Delta$. Therefore, due to the choice of the costs in M,

$$T(s_1)_\Delta T(W^2)_\Delta T(s_2)_\Delta T(W^3)_\Delta T(s_3)_\Delta T \cdots (W^l)_\Delta T(s_l)_\Delta T(W^r)_\Delta$$

has no C, and either $\overline{p_1} = (W)_\Delta T(s_1)_\Delta T p'(\Delta)$ or $\overline{p_1} = (W)_\Delta$. Suppose that $\overline{p_1} = (W)_\Delta T(s_1)_\Delta T p'(\Delta)$. In this case $\overline{p_2} = (W^2)_\Delta$. Since C^{2m} is a subsequence of $(W^2)_\Delta$, $\delta(u, (W^2)_\Delta) > t$. Therefore, $\overline{p_1} = (W)_\Delta$ and $k = 2$. It is not hard to check that in this case $u = Tu'T$ where u' is a string in alphabet $\{A, G\}$.

Since $u = Tu'T$, for every string $s_i \in S$, $\delta(u', s_i) \leqslant t$. In view of $\delta(A, G) = \delta(G, A) = m$, $d(u', s_i) \leqslant d$. Therefore, $s = u'$ is a solution for the original CS problem, which is a contradiction unless $\mathbf{P} = \mathbf{NP}$. Hence we have proved that AP is **NP**-complete. □

References

[1] M. Frances, A. Litman, On covering problems of codes, Theory Comput. Systems 30 (1997) 113–119.

[2] J.K. Lanctot, M. Li, B. Ma, S. Wang, L. Zhang, Distinguishing string selection problems, Proc. 10th ACM-SIAM Symp. on Discrete Algorithms, ACM-SIAM, New York, 1999, pp. 633–642.

[3] R.K. Moyzis, The human telomere, Sci. Amer. 265 (1991) 48–55.

[4] P.A. Pevzner, Multiple alignment, communication cost, and graph matching, SIAM J. Appl. Math. 52 (1992) 1763–1779.

[5] J.S. Sim, C.S. Iliopoulos, K. Park, W.F. Smyth, Approximate periods of strings, Theoret. Comput. Sci. 262 (2001) 557–568.